

JC949 U.S. PTO

10-23 00

H

Case Docket No. 65,135-008

Date: October 20, 2000

BOX PATENT APPLICATION
Assistant Commissioner for Patents
Washington, D. C. 20231

JC922 U.S. PTO
09/692852
10/20/00

Sir:

Transmitted herewith for filing is a Patent Application of:

Inventor(s): ANDREW R. OSBORN and MARTYN C. LORD

For : DISTRIBUTED MULTIPROCESSING SYSTEM

Enclosed are:

- ☒ 36 Page patent application.
- ☒ 9 Sheets of drawings -- ☒ Formal, ☐ Informal
- ☐ An Assignment of the invention to _____
- ☐ A Combined Declaration and Power of Attorney
- ☐ A Verified Statement By Assignee Claiming Small Entity Status
- ☐ An Associate Power of Attorney
- ☐ A Preliminary Amendment
- ☐ Information Disclosure Statement and PTO Form 1449 with copies of cited patents
- ☒ This application claims priority to U.S. Provisional Patent Application Serial No. _____, filed on October 18, 2000.

The filing fee has been calculated as shown below:

	No. Filed	No. Extra	Small Entity		Large Entity	
Basic Fee						\$ 710.00
Total Claims	67-20	47	@ \$ 9	-0-	@ \$ 18.00	846.00
Indep. Claims	4-3	1	@ 40	-0-	@ \$ 80.00	80.00
Multiple Depend. Claim(s) Present -0- \$ 260.00						
TOTAL \$ -0-				TOTAL \$ 1,636.00		

- ☐ Please charge my Deposit Account No. 08-2789 in the amount of \$ -0-. A duplicate copy of this sheet is enclosed.
- ☒ A check in the amount of \$ 1,636.00 to cover the filing fee is enclosed.
- ☐ The Commissioner is hereby authorized to charge payment of the following fees associated with this communication or credit any overpayment to Deposit Account No. 08-2789. A duplicate copy of this sheet is enclosed.
- ☒ Any additional filing fees required under 37 CFR § 1.16.
- ☒ Any patent application processing fees under 37 CFR § 1.17.

Respectfully Submitted,

HOWARD S. HOWARD

Samuel J. Haidle, Registration No. 42,619
The Pinchurst Office Center
39400 Woodward Avenue, Suite 101
Bloomfield Hills, MI 48304-5151
(248) 645-1483

CERTIFICATE OF MAILING BY "EXPRESS MAIL"

I hereby certify that the enclosed paper or fee is being deposited with the United States Postal Service as Express Mail, postage prepaid, in an envelope as "Express Mail Post Office to Addressee", Mailing Label No. EL 573 273 627 US and addressed to Box Patent Application, Assistant Commissioner for Patents, Washington, D.C. 20231, on October 20, 2000.

Brenda J. Hughes
Brenda J. Hughes

DISTRIBUTED MULTIPROCESSING SYSTEM

RELATED APPLICATION

This application claims priority to and all the benefits of United States Provisional
5 Patent Application Serial No. , filed on October 18, 2000 and entitled "Distributed
Multiprocessing System".

BACKGROUND OF THE INVENTION

1) TECHNICAL FIELD

10 The subject invention relates to a multiprocessing system which distributes data and
processes between a number of processors.

2) DESCRIPTION OF THE PRIOR ART

Data processing and distribution is utilized in a number of different manufacturing
15 and business related applications for accomplishing a virtually unlimited variety of tasks.
The systems implemented to accomplish these tasks utilize different design configurations
and are typically organized in a network fashion. Networks may be arranged in a variety of
configurations such as a bus or linear topology, a star topology, ring topology, and the like.

Within the network there are typically a plurality of nodes and communication links which
20 interconnect each of the nodes. The nodes may be computers, terminals, workstations,
actuators, data collectors, sensors, or the like. The nodes typically have a processor, a
memory, and various other hardware and software components. The nodes communicate
with each other over the communication links within the network to obtain and send
information.

A primary deficiency in the prior art systems is in the manner in which nodes communicate with other nodes. Currently, a first node will send a signal to a second node requesting information. The second node is already processing information such that the first node must wait for a response. The second node will at some time recognize the request by the first node and access the desired information. The second node then sends a response signal to the first node with the attached information. The second node maintains a copy of the information which it may need for its own processing purposes. The second node may also send a verification to ensure that the information data was received by the first node.

This type of communication may be acceptable in a number of applications where the time lost between the communications of the first and second nodes is acceptable. However, in many applications, such as real time compilation of data during vehicle testing, this lag time is unacceptable. Further, the redundancy in saving the same data in both the second and first nodes wastes memory space and delays processing time. Finally, the two way communication between the first and second nodes creates additional delays and the potential for data collision.

Accordingly, it would be desirable to have a data processing system which did not suffer from the deficiencies outlined above, is virtually seamless during the processing of data while reducing or eliminating unnecessary redundancies.

SUMMARY OF THE INVENTION AND ADVANTAGES

The subject invention overcomes the deficiencies in the prior art by providing a distributed multiprocessing system comprising a first processor for processing information at a first station and for assigning a first address to a first processed information. A second

processor processes information at a second station and assigns a second address to a second processed information. A central signal routing hub is interconnected between the first and second processors. Specifically, a first communication link interconnects the first processor and the hub for transmitting the first processed information between the first processor and the hub. A second communication link interconnects the second processor and the hub for transmitting the second processed information between the second processor and the hub. The central routing hub includes a sorter for receiving at least one of the first and second processed information from at least one of the first and second processors, thereby defining at least one sending processor. The hub and sorter also identify a destination of at least one of the first and second addresses of the first and second processed information, respectively. Finally, the hub and sorter send at least one of the first and second processed information without modification over at least one of the communication links to at least one of the first and second processors, thereby defining at least one addressed processor.

The subject invention also includes a method of communicating across the distributed multiprocessing system having the first processor and the second processor. The method comprising the steps of; processing information within at least one of the first and second processors; addressing the processed information; transmitting the processed information from at least one of the first and second processors across at least one of the communication links toward the hub, thereby defining at least one sending processor; receiving the processed information along with the address within the hub; identifying the destination of the address for the transmitted processed information within the hub; and sending the processed information without modification over at least one of the communication links to at least one of the first and second processors, thereby defining at

least one addressed processor.

In addition, the unique configuration of the subject invention may be practiced without the hub. In particular, first and second memory locations are connected to the first and second processors, respectfully, for storing received processed information. An
5 indexer is provided for indexing said first and second processors to define a different code for each of said processors for differentiating said processors. Further, said first and second processors each include virtual memory maps of each code such that said first and second processors can address and forward processed information to each of said indexed processors within said system.

10 The subject invention eliminating the hub also includes the steps of indexing the first and second processors to define a different code for each of the processors for differentiating the processors; creating a virtual memory map of each of the codes within each of the first and second processors such that the first and second processors can address and forward processed information to each of the indexed processors within the system;
15 and storing the processed information within the memory location of the addressed processor.

The subject invention therefore provides a data processing system which operates in a virtually instantaneous manner while reducing or eliminating unnecessary redundancies.

20 **BRIEF DESCRIPTION OF THE DRAWINGS**

Other advantages of the present invention will be readily appreciated as the same becomes better understood by reference to the following detailed description when considered in connection with the accompanying drawings wherein:

Figure 1 is a schematic view of the distributed multiprocessing system utilizing six

nodes interconnected to a single hub;

Figure 2 is another view of the system of Figure 1 illustrating possible paths of data flow between the nodes and the hub;

Figure 3 is a detailed schematic view of node 1 and node 2 as connected to the hub;

5 Figure 4 is a detailed schematic view of a memory space for node 1;

Figure 5 is a detailed schematic view of a processor for node 1;

Figure 6 is a detailed schematic view of a memory space for node 2;

Figure 7 is a detailed schematic view of a processor for node 2;

Figure 8 is an alternative embodiment illustrating only two nodes without a hub;

10 Figure 9 is a schematic view of two multiprocessing systems each having a hub with the hubs interconnected by a hub link;

Figure 10 is a schematic view of the two multiprocessing systems of Figure 8 before the hubs are interconnected;

15 Figure 11 is a schematic view of two multiprocessing systems each having a hub with the hubs interconnected by a common node;

Figure 12 is another schematic view of two multiprocessing systems interconnected by a common node;

Figure 13 is yet another schematic view of two multiprocessing systems interconnected by a common node;

20 Figure 14 is a schematic view of three multiprocessing systems each having a hub with the hubs interconnected by two common nodes;

Figure 15 is a schematic view of the system of Figure 1 illustrating another example of data flow between the nodes and the hub;

Figure 16 is a detailed schematic view of the processor and memory space of node 1

as node 1 processes information;

Figure 17 is a schematic view of the system of Figure 14 illustrating an incoming transmission of information;

Figure 18 is a schematic view of the system of Figure 14 illustrating an outgoing
5 transmission of information;

Figure 19 is a schematic view of the memory space of node 2 as the processed information of node 1 is stored into a real memory location of node 2;

Figure 20 is a schematic view of the system of Figure 1 illustrating yet another example of data flow between a node and the hub;

10 Figure 21 is a schematic view of the system of Figure 1 illustrating a incoming transmission from node 6;

Figure 22 is a schematic view of the system of Figure 20 illustrating a broadcast which sends outgoing transmissions to all nodes; and

Figure 23 is a schematic view of five systems interconnected by four common
15 nodes illustrating a broadcast through the system.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Referring to the Figures, wherein like numerals indicate like or corresponding parts throughout the several views, a distributed multiprocessing system is generally shown at 30
20 in Figure 1. The system 30 comprises a plurality of modules or nodes 1-6 interconnected by a central signal routing hub 32 to preferably create a star topology configuration. As illustrated, there are six nodes 1-6 connected to the hub 32 with each of the nodes 1-6 being indexed with a particular code. As an example of a code, numerical indicators 1 through 6 are illustrated. As appreciated, any suitable alpha/numeric indicator may be used to

differentiate one node from another. The shape, configuration, and orientation of the hub 32, which is shown as an octagon shape, is purely illustrative and may be altered to meet any desired need.

The nodes 1-6 may be part of a workstation or may be the workstation itself.

- 5 Illustrative of the versatility of the nodes 1-6, node 6 is part of a host computer 34, nodes 1, 2, 4, and 5 are connected to actuators 36 and node 3 is unconnected. It should be appreciated that the nodes 1-6 can be connected to any type of peripheral device or devices including multiple computers, actuators, hand held devices, and the like. For example, node 6 is shown also connected to a hand held device 35. Alternatively, none of the nodes
- 10 1-6 could be connected to a peripheral device which would create a completely virtual system.

- Referring also to Figure 2, the host computer 34 has a digital signal processing card 38 and preferably at least one peripheral device. The peripheral devices may be any suitable device as is known in the computer art such as a monitor, a printer, a key board, a
- 15 mouse, etc. As illustrated in Figure 2 and discussed in greater detail below, the nodes 1-6 preferably communicate with each other through the hub 32. For example, node 5 is shown communicating with node 6 through the hub 32 which in turn communicates with node 1 through the hub 32. Also, node 4 is shown communicating with node 3 through the hub 32.
- As discussed in greater detail below with respect to an alternative embodiment, when there
- 20 are only two nodes 1, 2 the hub 32 can be eliminated such that the nodes 1, 2 communicate directly with each other.

The subject invention is extremely versatile in the number of nodes which can be connected to the hub 32. There may be ten, one hundred, or thousands of nodes connected to the hub 32 or only a pair of nodes or even a single node connected to the hub 32. As will

be discussed in greater detail below, the nodes 1-6 can operate independently of each other.

In the preferred embodiment, the nodes 1-6 of the subject invention are utilized to compile data during a testing of a vehicle. In particular, during servo-hydraulic testing of a vehicle on a testing platform. Of course, the subject invention is in no way limited to this
5 envisioned application. The distributed multiprocessing system 30 of the subject invention can be used in virtually any industry to perform virtually any type of computer calculation or processing of data.

Referring to Figures 3 through 7, nodes 1 and 2 and the hub 32 are shown in greater detail. Each of the nodes 1-6 are virtually identical. Accordingly, nodes 3 through 6 can be
10 analogized as having substantially identical features illustrated in the detail of nodes 1 and 2. Each of the nodes 1-6 include a processor and a number of other components which will be outlined individually below.

The processors may be of different sizes and speeds. For example, node 6 may have a 1,500 MFbps processor and the remaining nodes may have a 300 MFbps processors.
15 The size and speed of the processor may be varied to satisfy a multitude of design criteria. Typically, the processor will only be of a size and speed to support the tasks or operation which are associated with the node 1-6. Further, the processors can be of different types which recognize different computer formats and languages.

Nodes 1 and 2 will now be discussed in greater detail. The first node, node 1,
20 includes a first processor 40 and the second node, node 2, includes a second processor 42. The first 40 and second 42 processors are indexed in concert with nodes 1 and 2 to define a different code for each of the processors 40, 42 for differentiating the processors 40, 42 in the same fashion as the nodes 1-6 are differentiated. In particular, an indexer 73, which is discussed in greater detail below, is included for indexing the first 40 and second 42

processors to define the different code for each of the processors 40, 42 for differentiating the processors 40, 42 and the nodes 1-6.

5 The first processor 40 processes information at a first station, i.e., node 1's location, and assigns a first address to a first processed information. Similarly, a second processor 42 processes information at a second station, i.e., node 2's location, and assigns a second address to a second processed information. As should be appreciated, the addresses are indexed to correlate to the indexing of the processors 40, 42 and the nodes 1-6.

10 First and second actuators 36 are connected to the first 40 and second 42 processors, respectively, for performing the testing operation during an operation of the system 30. There are additional components included within each of the nodes 1-6 such as a chipset 44 which interconnects the hub 32 and the processors 40, 42 and a buffer 46 disposed between each of the processors 40, 42 and the chipsets 44. Chipsets 44 were chosen for their transparent handling of data streams.

15 As shown in Figures 5 and 7, the first 40 and second 42 processors further include a hardware portion 48 for assigning the first and second addresses to the first and second processed information, respectively. In particular, the hardware portion 48 assigns a destination address onto the processed information indicative of the code of an addressed processor. The hardware portion 48 also conforms or rearranges the data or information to an appropriate format. As discussed above, the processors 40, 42 can be of different types
20 which recognize different computer formats. Hence, the hardware portion 48 ensures that the proper format is sent to the addressed processor. However, the addresses are preferably of a common format such that the hub 32 commonly recognizes these signals. Examples of the processors 40, 42 operation are discussed below in greater detail.

A first memory space 50 is connected to the first processor 40 and a second

memory space 52 is connected to the second processor 42. As shown in Figures 4 and 6, the first 50 and second 52 memory spaces are shown in greater detail, respectively. A first real memory location 54 is disposed within the first memory space 50 and is connected to the hardware portion 48 of the first processor 40. Similarly, a second real memory location 56 is disposed within the second memory space 52 and is connected to the hardware portion 48 of the second processor 42. During operation, the hardware portion 48 assigns a memory address onto the processed information indicative of the memory location of an addressed processor. The first 54 and second 56 real memory locations can therefore store received processed information, which is also discussed in greater detail below. The first 54 and second 56 real memory locations are not capable of reading the memory of another processor. In other words, the processors of a particular node 1-6 can read its own memory within its own memory locations but cannot read the memory stored within a memory location of another processor.

The first 54 and second 56 real memory locations may also have categorized message areas (not shown) such that multiple data inputs will not be overwritten. The categorized message areas could correlate to the memory addresses. In a similar fashion as above with regards to the processors 40, 42, the first 54 and second 56 real memory locations are of a size commensurate with the needs of the associated node 1-6.

Also illustrated within the first 50 and second 52 memory spaces at Figures 4 and 6, are first 58 and second 60 virtual memory maps. The first 40 and second 42 processors each include virtual memory maps 58, 60 of each code disposed within each of the first 40 and second 42 processors for each node 1-6 such that the first 40 and second 42 processors can address and forward processed information to each of the indexed processors within the system 30. The virtual memory maps 58, 60 are essentially a means for the processors 40,

42 to be able to address each other processor or node 1-6 within the system 30. The operation and specifics of the virtual memory maps 58, 60 will be discussed in greater detail below.

Referring back to Figures 5 and 7, each of the first 40 and second 42 processors further include at least one task 62. Each of the first 40 and second 42 processors will typically include a plurality of tasks 62 which can be performed in any order. A task 62 is a generic term for a specific operation or function being performed by a processor. The processors 40, 42 will include executable code for performing the tasks 62 which may be of different complexities. No one process or output associated with a task 62 is unique to any one node 1-6. In fact, many nodes 1-6 may have the same task 62 or tasks 62 for producing similar data.

As illustrated in the first processor 40 of node 1, there are four tasks 62 each occupying a different amount of space. A larger task space is meant to represent a task 62 which takes longer to process. The task 62 may be any suitable type of calculation, data collection, classification, or any other desired operation.

As also shown in Figures 5 and 7, each task 62 includes at least a pair of pointers 64, 66 for directing a flow of data from a sending processor to a destination processor. The pointers 64, 66 are illustrated as branching off of the fourth task 62 in Figure 5 and the third task 62 in Figure 7. As should be appreciated, there are pointers 64, 66 associated with each of the tasks 62 such that there is a continuous stream of information. Each pair of pointers 64, 66 includes a next task pointer 64 for directing the sending processor to a subsequent task 62 to be performed, and at least one data destination pointer 66 for sending the processed information to the hub 32. Preferably, there is only one next task pointer 64 such that there is a clear order of operation for the processors 40, 42. Conversely, there

may be any number of data destination pointers 66 such that the sending processor may simultaneously forward processed information to a multitude of addressed processors. Further, each of the processed information sent to the multitude of addressed processors may be different.

5 The next task 64 and data destination 66 pointers do not necessarily have to be operational for each task 62. For example, there may not be a need to send the particular information that the fourth task 62 has performed such that the data destination pointer 66 will not be operational. Conversely, the fourth task 62 may be the final task to be performed such that the next task pointer 64 will not be operational. Typically, at least one
10 of the pointers 64, 66 will be operational such that, at a minimum, the information will be sent to the hub 32 or a subsequent task 62 will be performed.

As shown back in Figures 1, 2, and 3, a first communication link 68 interconnects the first processor 40 of node 1 and the hub 32 for transmitting the first processed information between the first processor 40 and the hub 32. Similarly, a second
15 communication link 70 interconnects the second processor 42 of node 2 and the hub 32 for transmitting the second processed information between the second processor 42 and the hub 32. As appreciated, the hub 32 is capable of receiving processed information from all of the nodes 1-6 simultaneously and then forwarding the processed information to the correct destinations.

20 There are also communication links (not numbered) interconnecting each of the remaining processors of the remaining nodes 3-6 to the hub 32. As can be appreciated, the number of communication links is directly dependent upon the number of processors and nodes 1-6.

As discussed above, an indexer 73 is provided for indexing or organizing the first

40 and second 42 processors to define the different codes for each of the processors 40, 42, which differentiates the processors 40, 42 and the nodes 1-6. Preferably, the indexer 73 is disposed within the hub 32. Hence, when the nodes 1-6 are initially connected to the hub 32, the indexer 73 within the hub 32 begins to organize the nodes 1-6 in a particular order.

5 This is how the entire organization of the system 30 begins. The hub 32 and indexer 73 also create the mapping within the processors 40, 42 as part of this organization. As discussed above the mapping includes the first 58 and second 60 virtual memory maps of the first 40 and second 42 processors. The virtual memory maps 58, 60 outline each code disposed within each of the processors for each node 1-6 such that the processors can
10 address and forward processed information to each of the indexed processors within the system 30.

As shown in Figure 3, the central routing hub 32 includes a sorter 72 for receiving at least one of the first and second processed information from at least one of the first 40 and second 42 processors. By receiving the processed information, at least one sending
15 processor is defined. Each of the first 40 and second 42 processors may send processed information or only one of the first 40 and second 42 processors may send processed information. In any event, at least one of the first 40 and second 42 processors will be deemed as a sending processor.

The hub 32 and sorter 72 also identify a destination of at least one of the first and
20 second addresses of the first and second processed information, respectively. Finally, the hub 32 and sorter 72 send at least one of the first and second processed information without modification over at least one of the communication links 68, 70 to at least one of the first 40 and second 42 processors. The processor to which the information is being sent defines at least one addressed processor. The sorter 72 includes hardware 74 for determining the

destination addresses of the addressed processors.

As also shown in Figure 3, the first communication link 68 preferably includes first incoming 76 and first outgoing 78 transmission lines. Similarly, the second communication link 70 preferably includes second incoming 80 and second outgoing 82 transmission lines. The first 76 and second 80 incoming transmission lines interconnect the first 40 and second 42 processors, respectively, to the hub 32 for transmitting signals in only one direction from the first 40 and second 42 processors to the hub 32 to define a send-only system 30. Similarly, the first 78 and second 82 outgoing transmission lines interconnect the first 40 and second 42 processors, respectively, to the hub 32 for transmitting signals in only one direction from the hub 32 to the first 40 and second 42 processors to further define the send-only system 30. The chipsets 44 are designed to interconnect each of the incoming 76, 80 and outgoing 78, 82 transmission lines and the corresponding processors 40, 42 for creating a virtually transparent connection therebetween.

As will be discussed in greater detail below, the send-only system 30 eliminates the duplication of stored data. Preferably, the first 76 and second 80 incoming transmission lines and the first 78 and second 82 outgoing transmission lines are unidirectional optical fiber links. The optical fiber links are particularly advantageous in that the information is passed under high speeds and becomes substantially generic. Further, the unidirectional optical fiber links prevent the possibility of data collision. As appreciated, the first 76 and second 80 incoming and the first 78 and second 82 outgoing transmission lines may be of any suitable design without deviating from the scope of the subject invention.

The distributed multiprocessing system 30 can include any number of additional features for assisting in the uninterrupted flow of data through the system 30. For example,

a counter may be included to determine and control a number of times processed information is sent to an addressed processor. A sequencer may also be included to monitor and control a testing operation as performed by the system 30. In particular, the sequencer may be used to start the testing, perform the test, react appropriately to limits and events, establish that the test is complete, and switch off the test.

Referring to Figure 8, an alternative embodiment of the system 30 is shown wherein there are only two nodes 1, 2 and the hub 32 is eliminated. In this embodiment, a single communication link 68 interconnects the first processor 40 with the second processor 42 for transmitting the first and second processed information between the first 40 and second 42 processors. An indexer (not shown in this Figure) indexes the first 40 and second 42 processors to define a different code for each of the processors 40, 42 in a similarly manner as above. The first 40 and second 42 processors also each include virtual memory maps of each code such that the first 40 and second 42 processors can address and forward processed information to each other. There are also first 50 and second 52 memory locations for storing received processed information. The unique architecture allows the two nodes 1, 2 to communicate in a virtually seamless manner.

Specifically, the method of communicating between the first 40 and second 42 processors includes the steps of initially indexing the first 40 and second 42 processors to differentiate the processors 40, 42. Then the virtual memory maps of each of the codes is created within each of the first 40 and second 42 processors such that the first 40 and second 42 processors can address and forward processed information to each other. The processed information is transmitted by utilizing the virtual memory map of the sending processor, which may be from either node 1, 2, from the sending processor across the communication link toward the addressed processor, which is the corresponding opposite

node 1, 2. The processed information is then received along with the address in the addressed processor and the processed information is stored within the memory location of the addressed processor.

5 The remaining aspects of the nodes 1, 2 of this embodiment are virtually identical to the nodes 1, 2 of the primary embodiment. It should be appreciated that the details of the first 40 and second 42 processors as set forth in Figures 5 and 7, and the details of the first 50 and second 52 memory spaces as set forth in Figures 4 and 6 apply to this alternative embodiment.

10 Referring to Figure 9, a second hub 84, having nodes 7 and 8 with seventh and eighth processors, is interconnected to the first hub 32 by a hub link 86. The connection of one hub to another is known as cascading. As illustrated in Figure 10, the second hub 84, before connected to the first hub 32, indexed the two nodes 7 and 8 as node 1 and node 2. As should be appreciated, the nodes 1-8 of the two hubs 32, 84 must be re-indexed such that there are not two node 1s and node 2s.

15 Specifically, the indexer first indexes the first 32 and second 84 hubs to define a master hub 32 and secondary hub 84. In the illustrated example, hub number 1 is the master hub 32 and hub number 2 is the secondary hub 84. A key 88 is disposed within one of the first 32 and second 84 hubs to determine which of the hubs 32, 84 will be defined as the master hub. As illustrated, the key 88 is within the first hub 32. The indexer also
20 indexes the nodes 1-8 and processors to redefine the codes for each of the nodes 1-8 for differentiating the processors and nodes 1-8. When the first or master hub 32 is connected to the second or secondary hub 84 the entire virtual memory maps of each processor connected to the first hub 32 is effectively inserted into the virtual memory maps of each processor connected to the second hub 84 and vice versa. Hence, each hub 32, 84 can write

to all of the nodes 1-8 in the new combined or cascaded system 30 as shown in Figure 9.

Referring to Figures 11 through 13, there is illustrated various configurations for the combining of two hubs each having a plurality of nodes. These examples illustrate that the hubs can be attached through a node as opposed to utilizing the hub link 86. Further, as
5 shown in Figure 11, a node may be connected to more than one hub and the hubs may be connected to more than one common node.

Referring to Figure 14, there may be a third or more hubs interconnected to the system 30 through either a node (as shown) or by hub links 86. As can be appreciated, the versatility of the subject system 30 with regards to various combinations and configurations
10 of nodes and hubs is virtually limitless.

The particular method or steps of operation for communicating across the distributed multiprocessing system 30 is now discussed in greater detail. As above, the method will be further detailed with regards to communication between node 1 and node 2.

In particular, as illustrated in Figure 15, the given example is node 1 communicating to
15 node 2. It should be appreciated that the steps of operation will be substantially identical when communicating between any of the nodes 1-6 of the system 30 in any direction. Further, the nodes 1-6 can communicate directly with themselves as is discussed in another example below. In fact, a node 1-6 sending information to the hub 32 does not know the difference between writing to its own real memory location or the real memory location of
20 another node 1-6.

Referring to Figure 16, node 1 is shown again in greater detail. The method comprises the steps of processing information within at least one of the first 40 and second 42 processors. In this example the information is processed within the first processor 40 by proceeding through a number of tasks 62 in node 1. As discussed above, the tasks 62 may

be any suitable type of calculation, compilation or the like. Preferably, the processing of the information is further defined as creating data within the first processor 40. The creating of the data is further defined as compiling the data within the first processor 40. During the testing of the vehicle, which is discussed only as an illustrative embodiment, many of the processors of the nodes 1-6, including in this example node 1, will obtain and compile testing data.

To maintain the continuous flow of information, the system 30 further includes the step of directing the sending processor, which in this example is the first processor 40 of node 1, to a subsequent task 62 to be performed within the first processor 40 while simultaneously sending the processed information across one of the communication links 68, 70 to the hub 32. This step is accomplished by the use of the tasks 62 and pointers 64, 66. As shown, the first task 62 is first completed and then the first processor 40 proceeds to the second task 62. The pointers 64, 66 within the first task 62 direct the flow of the first processor 40 to the second task 62. Specifically, the data destination pointer 66 is silent and the next task pointer 64 indicates that the second task 62 should be the next task to be completed. The second task 62 is then completed and the first processor 40 proceeds to the fourth task 62. In this step, the next task pointer 64 of the second task 62 indicates to the first processor 40 that the fourth task 62 should be next, thereby skipping over the third task 62. The fourth task 62 is completed and the next task pointer 64 directs the flow to another task 62. The data destination pointer 66 of the fourth task 62 indicates that the information as processed after the fourth task 62 should be sent to the hub 32. The flow of information from the first task 62 to the second task 62 to the forth task 62 is purely illustrative and is in now way intended to limit the subject application.

The processed information from the fourth task 62 is then addressed and

transmitted from the first processor 40 across at least one of the communication links 68, 70 toward the hub 32. As discussed above, the communication links 68, 70 are preferably unidirectional. Hence, the step of transmitting the processed information is further defined as transmitting the processed information across the first incoming transmission line 76 in
5 only one direction from the first processor 40 to the hub 32 to define a send-only system 30. The transmitting of the processed information is also further defined by transmitting the data along with executable code from the sending processor to the addressed processor. As appreciated, the first 40 and second 42 processors initially do not have any processing capabilities. Hence, the executable code for the processors 40, 42 is preferably sent to the
10 processors 40, 42 over the same system 30. Typically, the executable code will include a command to instruct the processors 40, 42 to process the forwarded data in a certain fashion. It should also be noted that the transmitting of the processed information may be a command to rearrange or reorganize the pointers of the addressed processor. This in turn may change the order of the tasks which changes the processing of the addressed processor.
15 As appreciated, the transmitted processed data may include any combination of all or other like features.

The processed information is preferably addressed by the data destination pointer 66 directing the flow to the first virtual memory map 58 of node 1 and pointing to a destination node. The step of addressing the processed information is further defined as
20 assigning a destination address onto the processed information indicative of a code of an addressed processor. The step of addressing the processed information is further defined as assigning a memory address onto the processed information indicative of the memory location of the addressed processor, i.e., node 2. In this example the destination node, destination address, and memory address will be node 2 while the originating node will be

node 1.

The virtual memory map 58, 60 of each of the codes is created within each of the first 40 and second 42 processors such that the first 40 and second 42 processors can address and forward processed information to each of the indexed processors within the system 30. As discussed above, the virtual memory map 58, 60 is a means to which the processor can recognize and address each of the other processors in the system 30. By activating the data destination pointer 66 to send information to the hub 32, node 1 is then defined as a sending processor. As shown in Figure 16, the data destination pointer 66 directs the processed information to node 2 in the first virtual memory map 58 such that the destination address of node 2 will be assigned to this information.

Referring to Figure 17, the processed information is sent across the first incoming transmission line 76 of the first communication link 68. The processed information, along with the addresses, is then received within the hub 32.

Referring to Figure 18, the destination of the address for the transmitted processed information is identified within the hub 32 and the processed information is sent without modification over the second communication link 70 to, in this example, the second processor 42 of node 2. The step of sending the processed information without modification is further defined as sending the processed information over the second outgoing transmission line 82 in only one direction from the hub 32 to the second processor 42 to further define the send-only system 30. In this example, the hub 32 determines that the destination of the address is for node 2 which defines node 2 as an addressed processor with the destination address.

As shown in Figure 19, the processed information is then stored within the second real memory location 56 of the addressed second processor 42 wherein the second

processor 42 can utilize the information as needed. The processed information may be stored within the categorized message areas of the second real memory location 56 in accordance with the associated memory address. To save on memory space, the destination address (of node 2) may be stripped from sent processed information before the information
5 is stored in the second real memory location 56.

As also discussed above, the method of operation for the subject invention eliminates unnecessary duplication of information. When node 1 sends the processed information to the hub 32, which then travels to node 2, the information, which can include data, executable code, or both, is not saved at node 1 and is only stored at node 2. Node 2
10 does not send a confirmation and node 1 does not request a confirmation. Node 1 assumes that the information arrived at node 2. The subject system 30 is used to transport data to desired real memory locations where the data can be used during subsequent processing or evaluation.

The flow of communication across the system 30 will be precisely controlled such
15 that the nodes 1-6, i.e., node 2, will not receive unnecessary or processed information until it is needed. In other words, the processing at node 1 and the data destination pointer 66 at node 1 will be precisely timed to send the processed information across the system 30 to node 2 only moments before node 2 requires this information. Typically, node 2 will require the processed information of node 1 during its own processing of tasks. The system
20 30 of the subject invention is therefore virtually seamless and does not suffer from the deficiencies of requesting information from other nodes.

Another example of communicating across the subject system 30 is illustrated in Figure 20 wherein node 2 communicates with itself. The information is processed within the second processor 42 of node 2 by proceeding through a number of tasks 62. The

processed information is then addressed and transmitted from the second processor 42 across the second incoming transmission line 80 toward the hub 32. The processed information is addressed by the data destination pointer 66 directing the flow to the second virtual memory map 60 and pointing to the destination node. A destination address and a memory address are then assigned to the information. In this example the destination node, destination address, and memory address will be node 2 while the originating node will also be node 2. By activating the data destination pointer 66 to send information to the hub 32, node 2 is defined as a sending processor. The processed information, along with the address, is then received within the hub 32. The destination of the address for the transmitted processed information is identified within the hub 32 and the processed information is sent without modification over the second outgoing transmission line 82 to the designated processor. In this example, the hub 32 determines that the destination of the address is for node 2 which defines node 2 as an addressed processor with the destination address. The processed information is sent across the second outgoing transmission line 82 back to the second processor 42 within node 2. The processed information is then stored within the second real memory location 56 of the addressed second processor 42 of node 2. Node 2 has now successfully written information to itself.

By being able to write to themselves, the nodes 1-6 can perform self tests. The node, such as node 2 above, can send data and address the data using the second virtual memory space 60 and then later check to ensure that the data was actually received into the second real memory location 56 of node 2. This would test the hub 32 and communication link 68, 70 connections.

Referring to Figures 21 and 22, the system 30 also includes the step of simultaneously sending the processed information to all of the indexed processors by

simultaneously placing the destination addresses of each of the indexed processors onto the sent information. This is also know as broadcasting a message through the system 30. In the example shown in Figures 21 and 22, node 6 originates a message which is addressed to each of the nodes 1-6 in the system 30. The message or information is sent to the hub 32
5 across the associated incoming transmission line in the same manner as outlined above. The hub 32 determines that there are destination addresses for all of the nodes 1-6. This may be accomplished by choosing a special node number or I.D. which, if selected, automatically distributes the data to all nodes 1-6.

The message or information is then sent, without modification, across all of the
10 outgoing transmission lines to each of the nodes 1-6 as shown in Figure 22. The broadcasting is typically utilized for sending universally needed information, a shut down or start up message, an identify yourself message, or any like message or information.

Figure 23 illustrates the broadcasting of information from node 4 in a multi system 30, i.e., multi hub, configuration. The information is sent from node 4 to each hub in which
15 node 4 is connected. The hubs, which are shown as hub numbers 1, 2, and 3, in turn broadcast the information to each of their attached nodes 1-6. It should be appreciated, that a broadcast can be accomplished regardless of the configuration of the system 30.

The invention has been described in an illustrative manner, and it is to be understood that the terminology which has been used is intended to be in the nature of
20 words of description rather than of limitation. Obviously, many modifications and variations of the present invention are possible in light of the above teachings. It is, therefore, to be understood that within the scope of the appended claims the invention may be practiced otherwise than as specifically described.

CLAIMS

WHAT IS CLAIMED IS:

1. A method of communicating across a distributed multiprocessing system having a first processor and a second processor, the first and second processors being connected to a central signal routing hub by first and second communication links, respectively, said
5 method comprising the steps of;

processing information within at least one of the first and second processors;

addressing the processed information;

transmitting the processed information from at least one of the first and second
10 processors across at least one of the communication links toward the hub, thereby defining at least one sending processor;

receiving the processed information along with the address within the hub;

identifying the destination of the address for the transmitted processed information within the hub; and

15 sending the processed information without modification over at least one of the communication links to at least one of the first and second processors, thereby defining at least one addressed processor.

2. A method as set forth in claim 1 wherein the step of processing information is further defined as creating data within at least one of the first and second processors.

20 3. A method as set forth in claim 2 wherein the step of processing information is further defined as compiling the data within at least one of the first and second processors.

4. A method as set forth in claim 2 wherein the step of transmitting the processed information is further defined as transmitting data along with executable code from the sending processor to the addressed processor.

5. A method as set forth in claim 3 wherein the step of transmitting the processed information is further defined as transmitting the processed information across at least one of the communication links in only one direction from at least one of the first and second processors to the hub to define a send-only system.

5 6. A method as set forth in claim 5 wherein the step of sending the processed information without modification is further defined as sending the processed information over at least one of the communication links in only one direction from the hub to at least one of the first and second processors to further define the send-only system.

10 7. A method as set forth in claim 3 further including the step of directing the sending processor to a subsequent task to be performed within the processor while simultaneously sending the processed information across one of the communication links to the hub.

15 8. A method as set forth in claim 3 further including the step of indexing the first and second processors to define a different code for each of the processors for differentiating the processors.

9. A method as set forth in claim 8 further including the step of creating a virtual memory map of each of the codes within each of the first and second processors such that the first and second processors can address and forward processed information to each of the indexed processors within the system.

20 10. A method as set forth in claim 9 wherein the step of addressing the processed information is further defined as assigning a destination address onto the processed information indicative of the code of the addressed processor.

11. A method as set forth in claim 10 further including the step of storing the processed information within a memory location of the addressed processor.

12. A method as set forth in claim 11 wherein the step of addressing the processed information is further defined as assigning a memory address onto the processed information indicative of the memory location of the addressed processor.

5 13. A method as set forth in claim 12 further including the step of stripping the destination address from sent processed information before the information is stored in the memory location.

14. A method as set forth in claim 8 further including the step of interconnecting the hub to a second hub, having third and fourth processors, by a hub link.

10 15. A method as set forth in claim 14 further including the step of indexing the hubs to define a master hub and a secondary hub.

16. A method as set forth in claim 15 further including the step of indexing the first, second, third, and fourth processors in accordance with the master and secondary hub indexes to redefine the codes for each of the processors such that each of the processors can be differentiated.

15 17. A method as set forth in claim 16 further including the step of simultaneously sending the processed information to all of the indexed processors by simultaneously placing the destination addresses of each of the indexed processors onto the sent information.

20 18. A method as set forth in claim 17 further including the step of limiting the number of times that the processed information can be sent form a sending processor.

19. A distributed multiprocessing system comprising;

a first processor for processing information at a first station and for assigning a first address to a first processed information,

a second processor for processing information at a second station and for assigning

5 a second address to a second processed information,

a central signal routing hub,

a first communication link interconnecting said first processor and said hub for transmitting said first processed information between said first processor and said hub,

a second communication link interconnecting said second processor and said hub

10 for transmitting said second processed information between said second processor and said hub,

said central routing hub including a sorter for receiving at least one of said first and second processed information from at least one of said first and second processors, thereby defining at least one sending processor. and for identifying a destination of at least one of

15 said first and second addresses of said first and second processed information, respectively, and for sending at least one of said first and second processed information without modification over at least one of said communication links to at least one of said first and second processors, thereby defining at least one addressed processor.

20. A system as set forth in claim 19 wherein said first communication link

20 includes first incoming and first outgoing transmission lines.

21. A system as set forth in claim 20 wherein said second communication link includes second incoming and second outgoing transmission lines.

22. A system as set forth in claim 21 wherein said first and second incoming transmission lines interconnect said first and second processors, respectively, to said hub

for transmitting signals in only one direction from said first and second processors to said hub to define a send-only system.

23. A system as set forth in claim 22 wherein said first and second outgoing transmission lines interconnect said first and second processors, respectively, to said hub
5 for transmitting signals in only one direction from said hub to said first and second processors to further define said send-only system.

24. A system as set forth in claim 23 wherein said first and second incoming transmission lines and said first and second outgoing transmission lines are unidirectional optical fiber links.

10 25. A system as set forth in claim 19 further including at least one actuator connected to at least one of said first and second processors, respectively, for performing a testing operation during an operation of said system.

26. A system as set forth in claim 25 wherein said actuator is further defined as servo-hydraulic actuator.

15 27. A system as set forth in claim 19 further including an indexer for indexing said first and second processors to define a different code for each of said processors for differentiating said processors.

28. A system as set forth in claim 27 wherein said first and second processors each include virtual memory maps of each code such that said first and second processors can
20 address and forward processed information to each of said indexed processors within said system.

29. A system as set forth in claim 28 wherein each of said first and second processors further include a hardware portion for assigning said first and second addresses to said first and second processed information, respectively.

30. A system as set forth in claim 29 wherein said hardware portion assigns a destination address onto said processed information indicative of said code of said addressed processor.

5 31. A system as set forth in claim 30 further including first memory location connected to said hardware portion of said first processor and second memory location connected to said hardware portion of said second processor, said first and second memory locations storing received processed information.

10 32. A system as set forth in claim 31 wherein said hardware portion assigns a memory address onto said processed information indicative of an associated first and second memory location of said addressed processor.

33. A system as set forth in claim 27 further including a second hub, having third and fourth processors, interconnected to said first hub by a hub link.

15 34. A system as set forth in claim 33 wherein said indexer indexes said first and second hubs to define a master hub and secondary hub and indexes said first, second, third, and fourth processors to redefine said codes for each of said processors for differentiating said processors.

35. A system as set forth in claim 34 further including a key disposed within one of said first and second hubs to determine which of said hubs will be defined as said master hub.

20 36. A system as set forth in claim 19 wherein each of said first and second processors further include at least one task.

37. A system as set forth in claim 36 wherein said processors include executable code for processing information defined by each of said tasks.

38. A system as set forth in claim 37 wherein said task includes at least a pair of

pointers for directing a flow of data from said sending processor to said destination processor.

39. A system as set forth in claim 38 wherein said pair of pointers includes a next task pointer for directing said sending processor to a subsequent task to be performed, and
5 at least one data destination pointer for sending said processed information across one of said incoming transmission lines to said hub.

40. A system as set forth in claim 39 wherein said at least one data destination pointer includes a plurality of data destination pointers to simultaneously forward processed information to a plurality of addressed processors.

10 41. A system as set forth in claim 39 further including a chipset interconnected between each of said incoming and outgoing communication links and said corresponding processors for creating a virtually transparent connection there between.

42. A system as set forth in claim 41 further including a buffer disposed between each of said processors and said chipsets.

15 43. A system as set forth in claim 42 further including a counter for determining a number of times said processed information is sent to said addressed processor.

44. A system as set forth in claim 43 further including a sequencer for monitoring and controlling a testing operation as performed by said system.

20 45. A system as set forth in claim 19 further including a host computer connected to one of said first and second processors, said host computer having a digital signal processing card and at least one peripheral device.

46. A system as set forth in claim 45 wherein said peripheral devices are further defined as a monitor, a printer, a key board, and a mouse.

47. A system as set forth in claim 19 wherein said sorter includes hardware for determining said destination addresses of said addressed processors.

48. A method of communicating across a distributed multiprocessing system having at least a first processor and a second processor, the first and second processors each having a memory location and being connected to each other by a communication link, said method comprising the steps of;

5 indexing the first and second processors to define a different code for each of the processors for differentiating the processors;

creating a virtual memory map of each of the codes within each of the first and second processors such that the first and second processors can address and forward processed information to each other;

10 processing information within at least one of the first and second processors to define at least one of the first and second processors as a sending processor;

addressing the processed information to define at least one of the first and second processors as an addressed processor;

15 transmitting the processed information by utilizing the virtual memory map of the sending processor from the sending processor across the communication link toward the addressed processor;

receiving the processed information along with the address in the addressed processor; and

20 storing the processed information within the memory location of the addressed processor.

49. A method as set forth in claim 48 wherein the step of processing information is further defined as creating data within at least one of the first and second processors.

50. A method as set forth in claim 49 wherein the step of processing information is further defined as compiling the data within at least one of the first and second processors.

51. A method as set forth in claim 49 wherein the step of transmitting the processed information is further defined as transmitting data along with executable code from the sending processor to the addressed processor.

52. A method as set forth in claim 50 wherein the step of transmitting the
5 processed information is further defined as transmitting the processed information across the communication link in only one direction from at least one of the first and second processors to the other of the first and second processors to define a send-only system.

53. A method as set forth in claim 50 further including the step of directing the sending processor to a subsequent task to be performed within the processor while
10 simultaneously sending the processed information across the communication link to the respective first and second processor.

54. A method as set forth in claim 50 wherein the step of addressing the processed information is further defined as assigning a destination address onto the processed information indicative of the code of the addressed processor.

55. A method as set forth in claim 54 wherein the step of addressing the processed
15 information is further defined as assigning a memory address onto the processed information indicative of the memory location of the addressed processor.

56. A method as set forth in claim 55 further including the step of stripping the destination address from sent processed information before the information is stored in the
20 memory location.

57. A method as set forth in claim 55 further including the step of simultaneously sending the processed information to all of the indexed processors by simultaneously placing the destination addresses of each of the indexed processors onto the sent information.

58. A distributed multiprocessing system comprising;

a first processor for processing information at a first station and for assigning a first address to a first processed information,

a first memory location connected to said first processor for storing processed
5 information,

a second processor for processing information at a second station and for assigning a second address to a second processed information,

a second memory location connected to said second processor for storing processed information,

10 a communication link interconnecting said first processor with said second processor for transmitting said first and second processed information between said first and second processors,

an indexer for indexing said first and second processors to define a different code for each of said processors for differentiating said processors,

15 said first and second processors each including virtual memory maps of each code such that said first and second processors can address and forward processed information to each other, thereby defining at least one sending processor, and said first and second memory locations storing received processed information to define at least one addressed processor.

20 59. A system as set forth in claim 58 wherein said communication link includes incoming and outgoing transmission lines for transmitting signals in only one direction between said first and second processors to define a send-only system.

60. A system as set forth in claim 59 wherein each of said first and second processors further include a hardware portion for assigning said first and second addresses

to said first and second processed information, respectively.

61. A system as set forth in claim 60 wherein said hardware portion assigns a destination address onto said processed information indicative of said code of said addressed processor.

5 62. A system as set forth in claim 61 wherein said hardware portion assigns a memory address onto the processed information indicative of the memory location of the addressed processor.

63. A system as set forth in claim 58 wherein each of said first and second processors further include at least one task.

10 64. A system as set forth in claim 63 wherein said processors include executable code for processing information defined by each of said tasks.

65. A system as set forth in claim 64 wherein said task includes at least a pair of pointers for directing a flow of data from said sending processor to said destination processor.

15 66. A system as set forth in claim 65 wherein said pair of pointers includes a next task pointer for directing said sending processor to a subsequent task to be performed, and a data destination pointer for sending said processed information across one of said incoming transmission lines to said addressed processor.

20 67. A system as set forth in claim 66 wherein said at least one data destination pointer includes a plurality of data destination pointers to simultaneously forward processed information to a plurality of addressed processors.

ABSTRACT OF THE DISCLOSURE

A distributed multiprocessing system includes a number of nodes 1-6 interconnected through a central signal routing hub. Each of the nodes 1-6 are preferably connected to an actuator and include a processor for processing information. The nodes
5 1-6 also assign addresses to the processed information. Communication links interconnect the processors with the hub for transmitting the processed information between the processors and the hub. The central routing hub includes a sorter for receiving processed information from the processors. The hub and sorter identify a destination of the processed information and send the processed information without
10 modification over an associated communication link to an addressed processor. The system of the subject invention creates a virtually seamless stream of data for real time compilation of information during a testing of a vehicle.

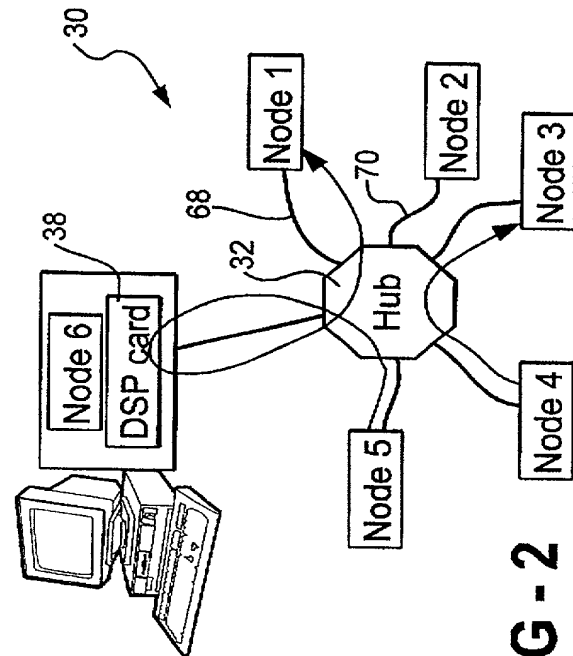


FIG - 1

FIG - 2

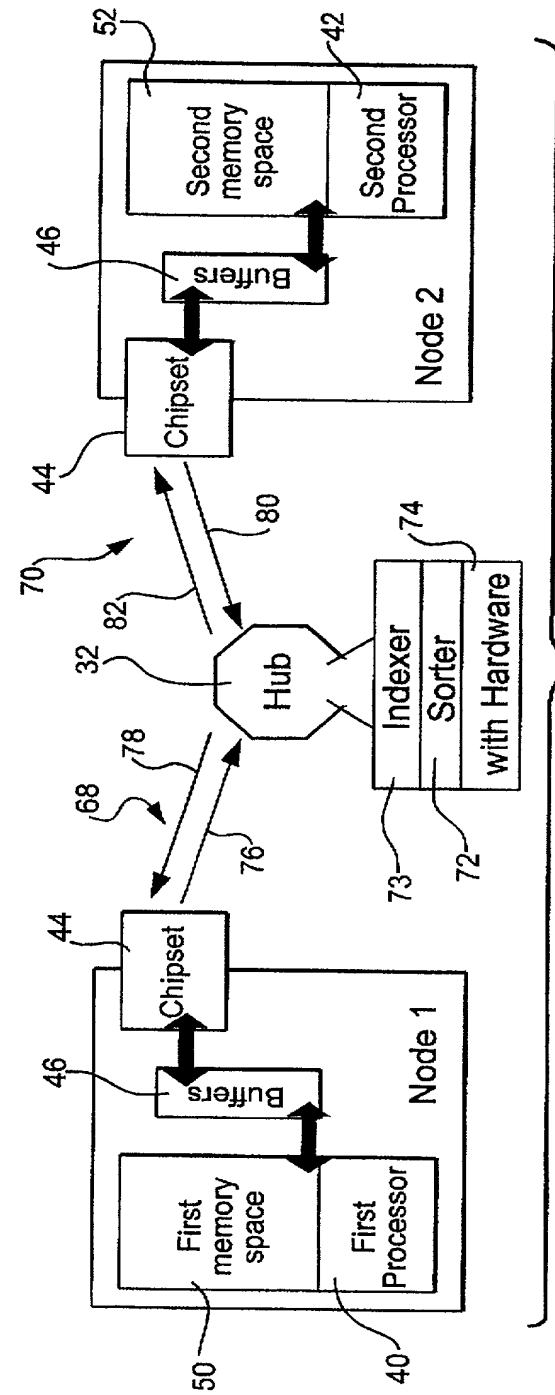


FIG - 3

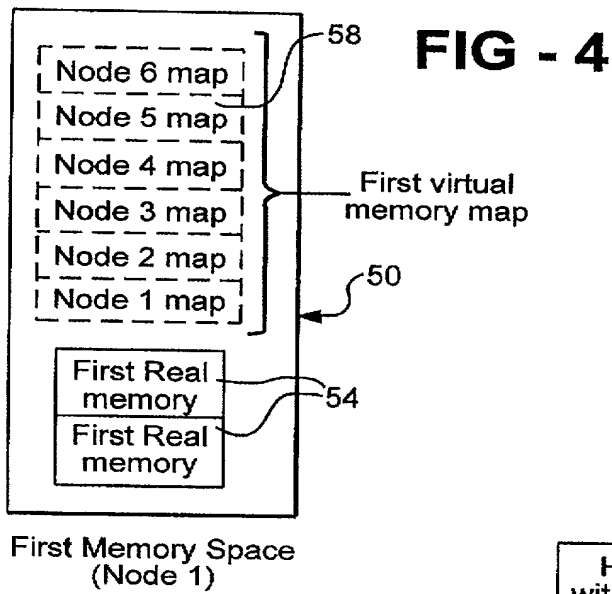
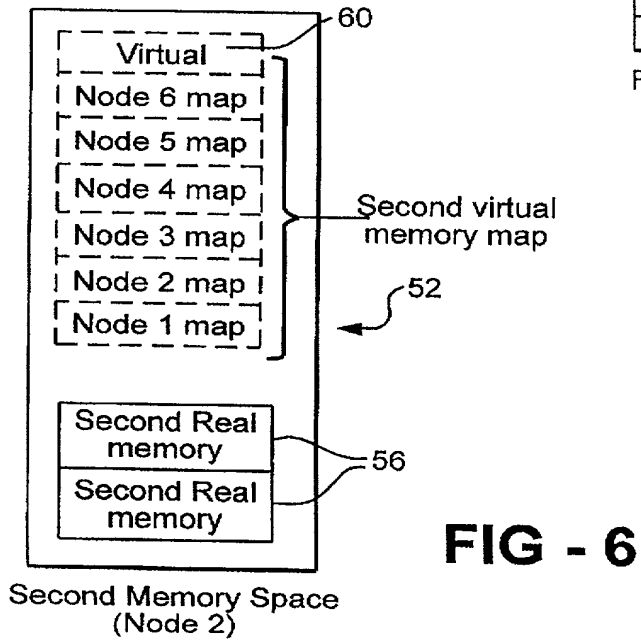
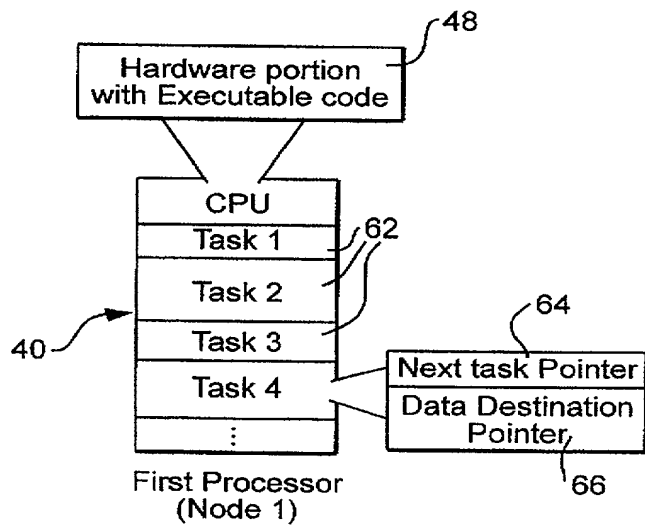


FIG - 5



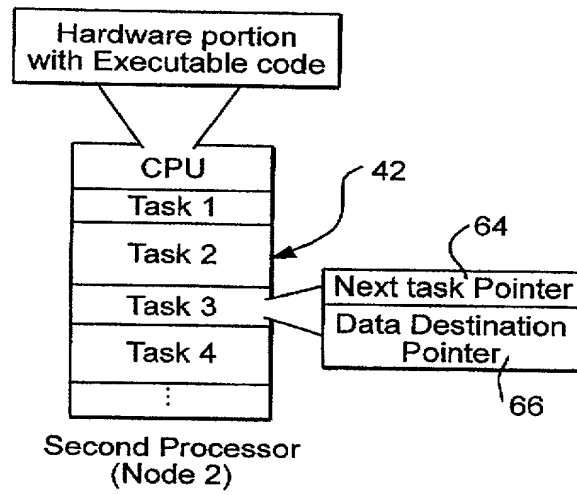


FIG - 7

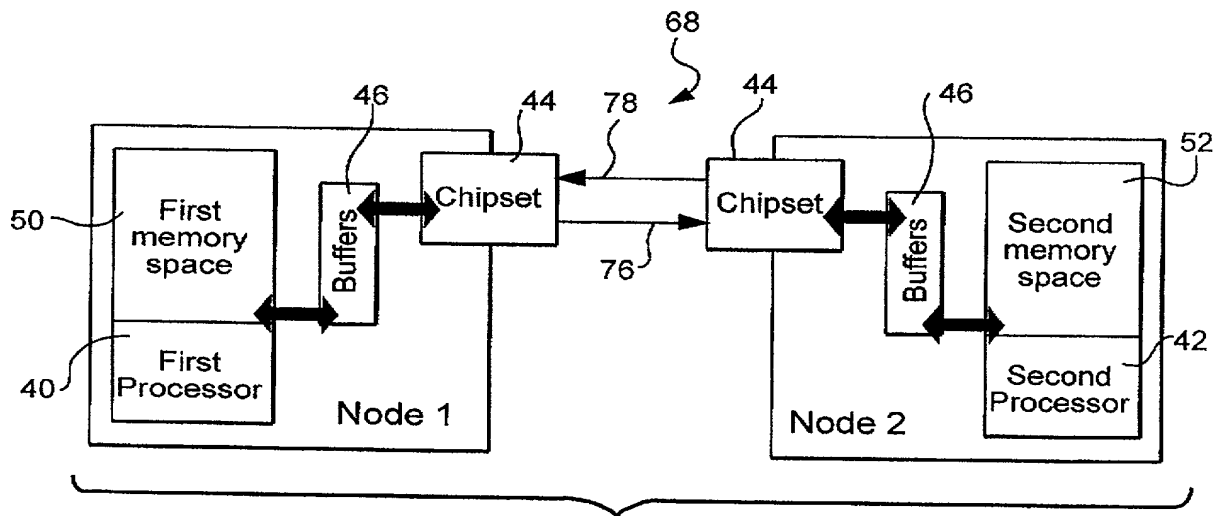


FIG - 8

FIG - 9

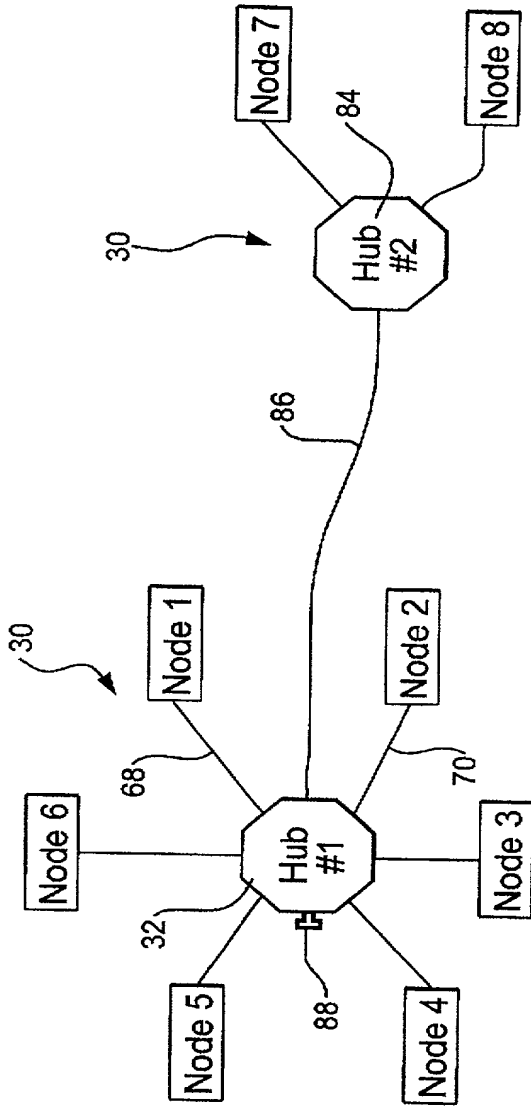
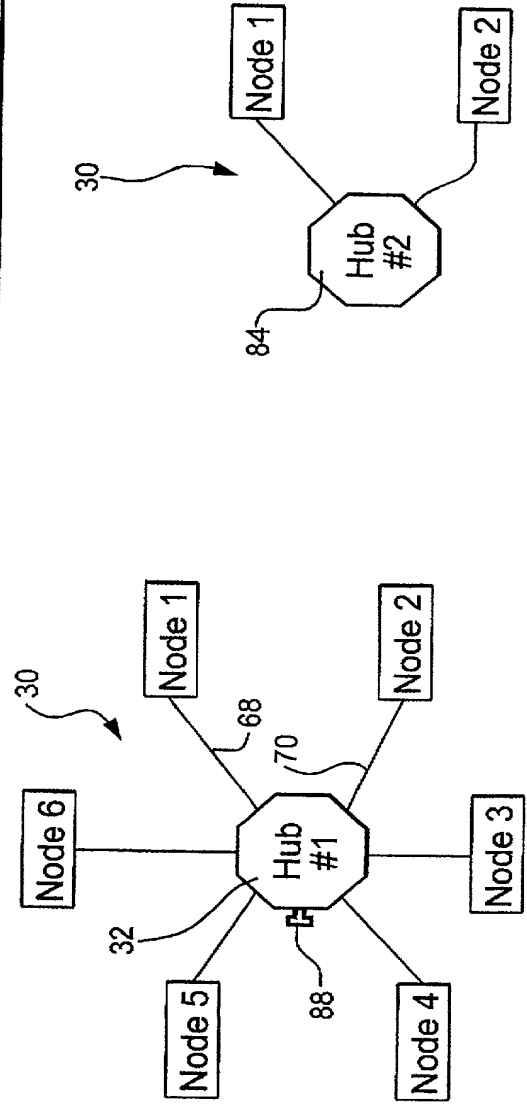


FIG - 10



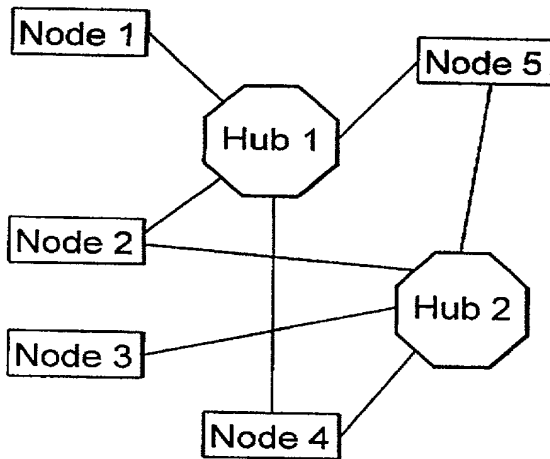


FIG - 11

FIG - 12

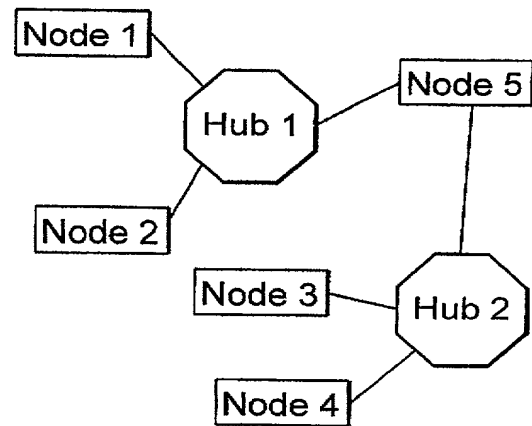
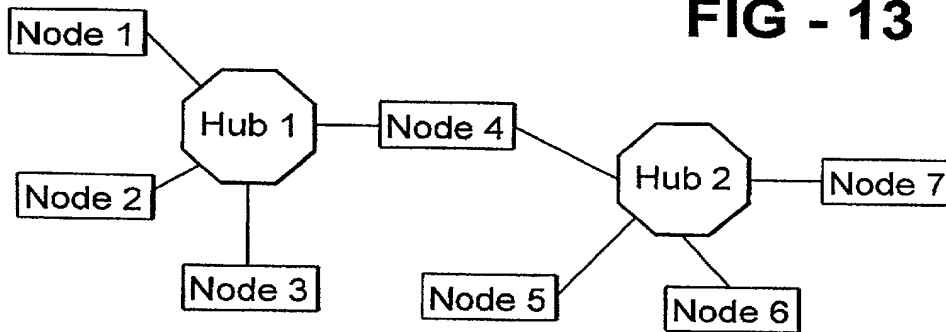


FIG - 13



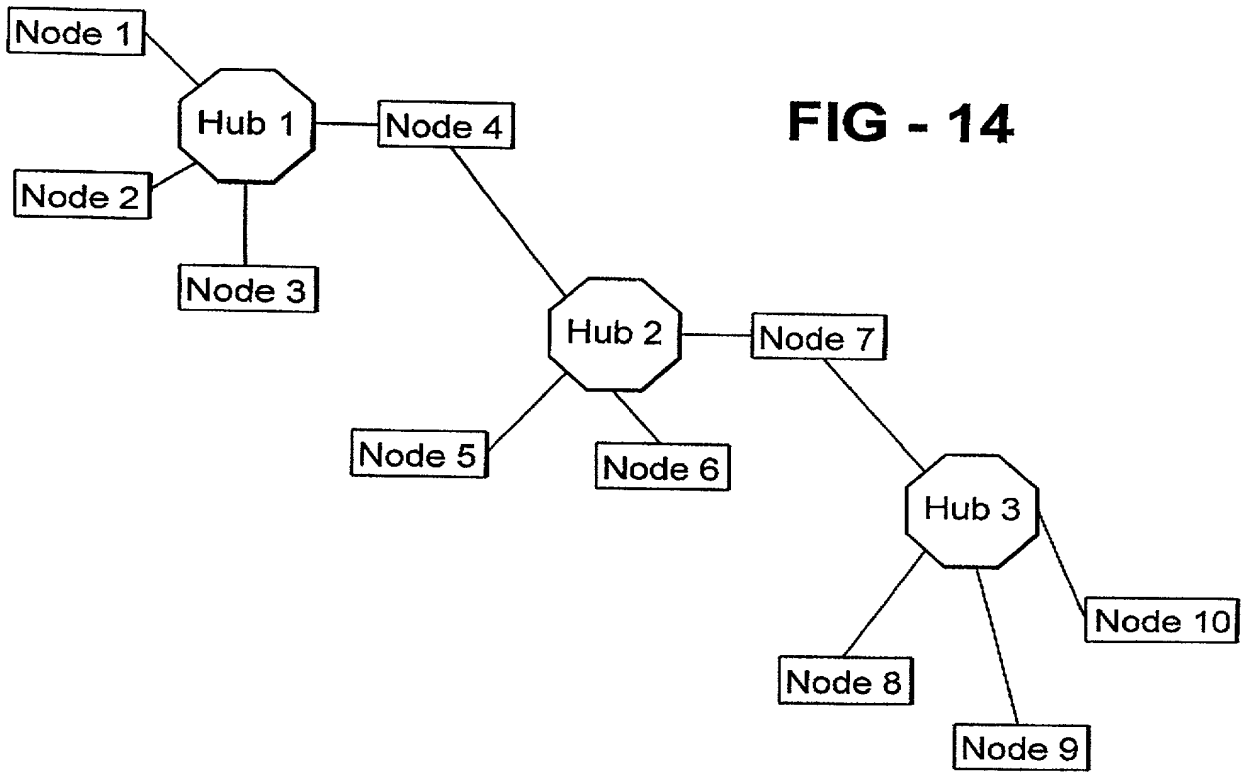


FIG - 15

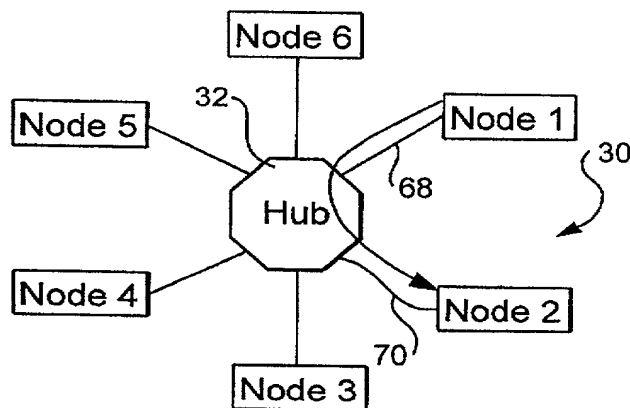


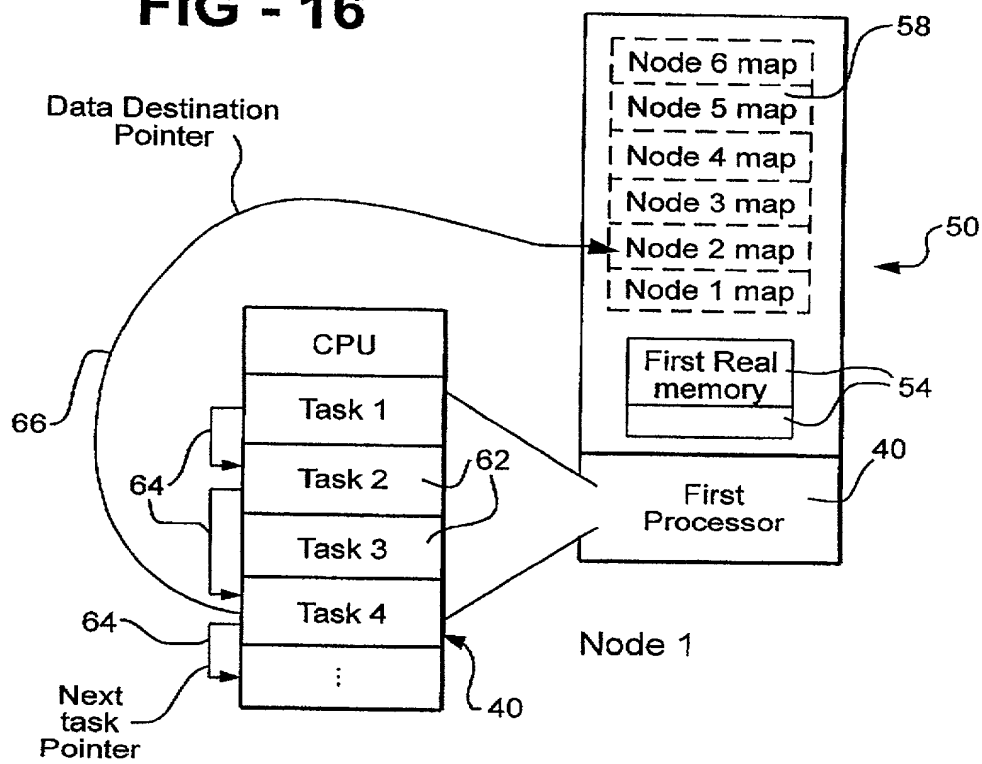
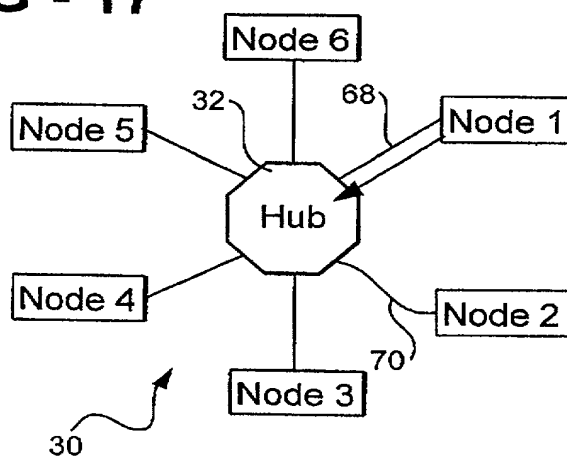
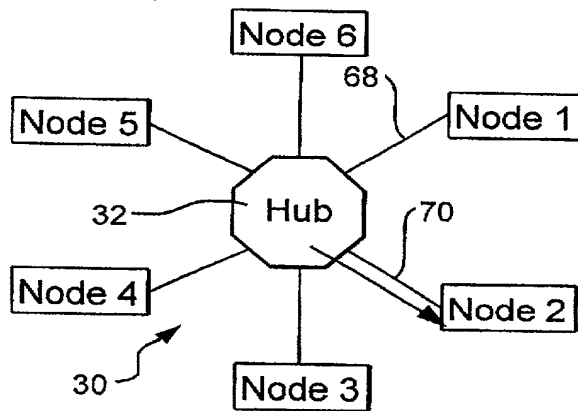
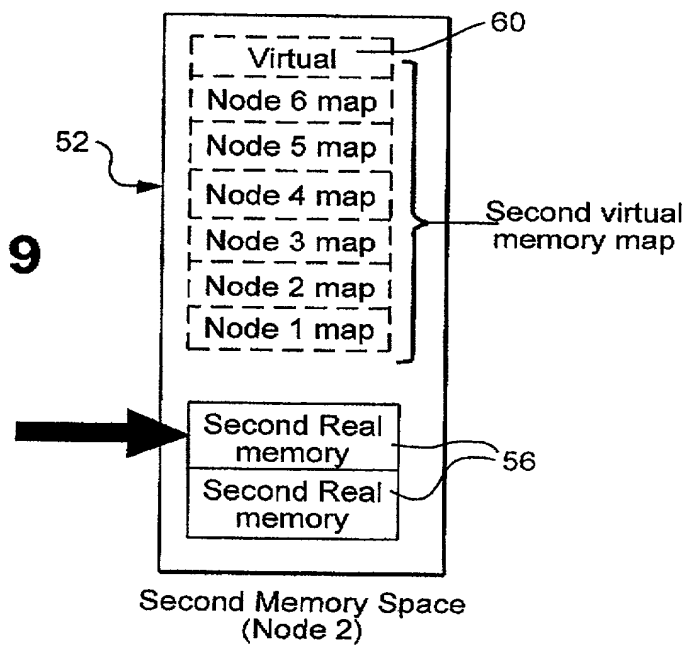
FIG - 16**FIG - 17**

FIG - 18**FIG - 19****FIG - 20**